

Restricted Boltzmann Machines with Gaussian Visible Units Guided by Pairwise Constraints

Jielei Chu, Hongjun Wang*, Hua Meng, Peng Jin and Tianrui Li, *Senior member, IEEE*



Abstract—Restricted Boltzmann machines (RBMs) and their variants are usually trained by contrastive divergence (CD) learning, but the training procedure is an unsupervised learning approach, without any guidances of the background knowledge. To enhance the expression ability of traditional RBMs, in this paper, we propose pairwise constraints restricted Boltzmann machine with Gaussian visible units (pcGRBM) model, in which the learning procedure is guided by pairwise constraints and the process of encoding is conducted under these guidances. The pairwise constraints are encoded in hidden layer features of pcGRBM. Then, some pairwise hidden features of pcGRBM flock together and another part of them are separated by the guidances. In order to deal with real-valued data, the binary visible units are replaced by linear units with Gaussian noise in the pcGRBM model. In the learning process of pcGRBM, the pairwise constraints are iterated transitions between visible and hidden units during CD learning procedure. Then, the proposed model is inferred by approximative gradient descent method and the corresponding learning algorithm is designed in this paper. In order to compare the availability of pcGRBM and traditional RBMs with Gaussian visible units, the features of the pcGRBM and RBMs hidden layer are used as input ‘data’ for K-means, spectral clustering (SP) and affinity propagation (AP) algorithms, respectively. A thorough experimental evaluation is performed with sixteen image datasets of Microsoft Research Asia Multimedia (MSRA-MM). The experimental results show that the clustering performance of K-means, SP and AP algorithms based on pcGRBM model are significantly better than traditional RBMs. In addition, the pcGRBM model for clustering task shows better performance than some semi-supervised clustering algorithms.

Index Terms—restricted Boltzmann machine (RBM); pairwise constraints; contrastive divergence (CD); unsupervised clustering; semi-supervised clustering.

1 INTRODUCTION

Hinton and Sejnowski[1] proposed a learning algorithm for general Boltzmann machine which has hidden-to-hidden and visible-to-visible connections, but in practice it was too slow to be used. Then, the restricted Boltzmann machine (RBM) was proposed by[2] in 1986, which has no lateral connections among nodes in each layer, so the learning procedure becomes much more efficient than general Boltzmann machine. There has been extensive research into the RBM since Hinton proposed fast learning algorithms[3],

[4] by contrastive divergence (CD) learning algorithm. Several power and tractability deep networks was proposed, including deep belief networks[5], deep autoencoder[6], deep Boltzmann machine[7], deep dropout neural net[8]. Until now, a large number of successful applications built on the RBMs have appeared, e.g., classification[9], [10], [11], [12], [13], feature learning[14], facial recognition[15], collaborative filtering[16], topic modelling[17], speech recognition[18], natural language understanding[19], computer vision[20], dimensionality reduction[21], voice conversion[22], musical genre categorization[23], real-time key point recognition[24] and periocular recognition[25].

The classic RBM has great ability of extracting hidden features from original data. More and more researchers proposed variant RBMs and their deep networks which were based on classic RBM, e.g., fuzzy restricted Boltzmann machine(FRBM)[26], classification RBM[27], spike-and-slab restricted Boltzmann machine (ssRBM)[28], Gaussian restricted Boltzmann machines (GRBMs)[29], sparse restricted Boltzmann machine (SRBM)[30], over-replicated softmax model[31], temporal restricted Boltzmann machines (RTRBMs)[32], circle convolutional restricted Boltzmann machine (CCRBM)[33], adaptive restricted Boltzmann machine[34], relevance restricted Boltzmann machine (ReRBM)[35], theta-restricted Boltzmann machine (theta-RBM)[36], disjunctive factored four-way conditional restricted Boltzmann machine (DFFW-CRBM)[37], centered convolutional restricted Boltzmann machines (CCRBM)[38], social restricted Boltzmann machine (SRBM)[39], temperature based restricted Boltzmann machines (TRBMs)[40] and deep feature coding architecture[41].

However, since the learning procedures of classic RBM and its variants are unsupervised methods, their processes of feature extraction are non-directional and conducted under no guidance. To remedy these weakness, this paper proposes a pairwise constraints restricted Boltzmann machine with Gaussian visible units (pcGRBM) and corresponding learning algorithm, where the learning procedure is guided by pairwise constraints which come from labels. In pcGRBM model, the pairwise constraints which is instance-level prior knowledge guide the process of encoding, some pairwise hidden features of pcGRBM flock together and another part of them are separated by the guidances, then the process of feature extraction is no longer non-directional. Then, the background knowledge of instance-level pairwise constraints are encoded in hidden

Jielei Chu, Hongjun Wang, Hua Meng and Tianrui Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China. e-mail: jieleichu@home.swjtu.edu.cn, wanghongjun@swjtu.edu.cn, huameng@swjtu.edu.cn, trli@swjtu.edu.cn. Peng Jin is with the School of Computer Science, Leshan Normal University, 614000, Leshan, China. e-mail: jandp@pku.edu.cn

layer features of pcGRBM. In order to testify the availability of pcGRBM, we design three structures of clustering, in which the features of the hidden layer of the pcRBM are used as input ‘data’ for unsupervised clustering algorithms. The experimental results show that the clustering performance of K-means, SP and AP algorithms based on pcGRBM model are significantly better than traditional RBMs. In addition, the pcGRBM model for clustering is better performance than some semi-supervised algorithms (Cop-Kmeans[42], Semi-Spectral clustering (Semi-SP)[43] and semi-supervised affinity propagation (Semi-AP)[44]).

The remainder of this paper is organized as follows. In the next section, we outline the related work and provide the preliminary in section III, which includes pairwise constraints, RBM and Gauss visible units. The proposed pcGRBM model and its learning algorithm are introduced in section IV. Next, the remarkable performance of the pcGRBM model is affirmed by the task of clustering on MSRA-MM in section V. Finally, Section VI summarizes our contributions.

2 RELATED WORK

Due to the outstanding performance, more and more variants of RBM have been proposed by researchers. There are several common methods to develop standard RBM such as adding connections information between the visible units and the hidden units, changing the value type of visible or hidden units, expanding the relationships of the units between visible layer and hidden layer from constant to variable by fuzzy mathematics, constructing deep network based on autoencoder[21] by pairwise constraints.

To add connections information between the visible units into RBM is a kind of methods for developing standard RBM. Osindero and Hinton proposed a semi-restricted Boltzmann machines (SRBM)[45] which has lateral connections between the visible units, but these lateral connections are unit-level semi-supervised information. The learning procedure includes two stages: the first one is the visible to hidden connections which is same as a classic RBM and the second one is the lateral connections which is applied the same learning procedure as the first one. In order to enforce hidden units to be pairwise uncorrelated and to maximize entropy, Tomczak[46] proposed to add penalty term to the log-likelihood function. His framework of learning informative features is unit-level pairwise and for classification problem, while our model is instance-level pairwise and for clustering task. Zhang et al.[47] built deep belief network based on SRBM for classification. Given the hidden units, the visible units of the SRBM form a Markov random field. However, the main weakness of the SRBM is that there are massive parameters for high-dimensional data, if every pairs of visible units have relations. Sutskever and Hinton proposed temporal restricted Boltzmann machine (TRBM)[48] by adding directed connections between previous and current states of the visible and hidden units. There are three kinds of connections of the full TRBM, e.g., connections between the visible units, connections between the hidden and visible units and connections between the hidden units. Furthermore, they proposed the recurrent TRBM

(RTRBM)[32]. It is easy to compute the gradient of the log-likelihood and infer exactly. Mnih and Hinton proposed the conditional restricted Boltzmann machines (CRBMs)[49] by adding conditioning vector which determines increments to the biases of the visible and hidden layer of the traditional RBM.

By changing hidden units with relevancy is another kind of methods for developing standard RBM. Courville et al.[28] developed the spike-and-slab restricted Boltzmann machine (ssRBM). The ssRBM is defined as having each hidden unit associated with the product of a binary “spike” latent variable and a real-valued “slab” latent variable. In order to keep learning efficiency, as a model of natural images, the binary hidden units of the ssRBM maintain the simple conditional independence structure when they encode the conditional covariance of visible units by exploiting real-valued slab variables.

In general, the relationships of the units between the visible layer and the hidden layer are restricted to be constants. In order to break through this restrictions, Chen et al.[26] proposed a fuzzy restricted Boltzmann machine (FRBM) to enhance deep learning capability which can avoid the flaw. The FRBM model parameters are replaced by fuzzy numbers and the regular RBM energy function is given by fuzzy free energy functions. Moreover, the deep networks are designed by the fuzzy RBMs to boost deep learning. Nie et al.[20] proposed to theoretically extend the conventional RBMs by introducing another term in the energy function to explicitly model the local spatial interactions in the input data.

Conventional RBM defines the units of visible and hidden layer to be binary, but this limitation cannot meet the needs in practice. Then one common way is to replace them by means of Gaussian linear units, that is Gaussian-Bernoulli restricted Boltzmann machines (GBRBMs)[50]. The GBRBMs have the ability to learn meaningful features both in modeling natural images and in a two-dimensional separation task. But, as we know, it is difficult to learn the GBRBMs. So, Cho et al.[51] proposed a novel method to improve their learning efficiency. The new method includes three parts, e.g., changing energy function by different parameterizations to facilitate learning, parallel tempering learning and adaptive learning rate. Moreover, the deep networks of Gaussian-Bernoulli deep Boltzmann machine (GDBM)[52], [53] has been developed by the GBRBM in recent years. The GDBM is designed by adding multiple layer of hidden units and applied to continuous data.

Furthermore, Zhang et al. proposed a mixed model named as supervision guided autoencoder (SUGAR)[54] which includes three components: main network, auxiliary network and bridge. The main network is a sparsity-encouraging variant of the autoencoder[21], that is the unsupervised autoencoder. The auxiliary network is constructed by pairwise constraints, that is the supervised learning. The two heterogeneous networks are designed and each of which encodes either unsupervised or supervised data structure respectively. The main network and auxiliary network are connected by the bridge which is used to enforce the correlation of the parameters. Comparing SUGAR with supervised learning and supervised deep networks, it

has flexible utilization of supervised information and better balances the numerical tractability.

In the work of [55], Chen proposed a deep network structure based on RBMs which is the most related to our work. Both the work of [55] and our work aim to solve the similar problems, e.g., how to obtain suitable features for clustering by non-linear mapping and how to use pairwise constraints during learning process, but the model and the solution are different. They use RBMs to initialize connection weights with CD learning, learning process is still unsupervised method, then the learned weights are used to incorporate pairwise constraints in features space by maximum margin techniques. However, our pcGRBM model is based on RBMs with Gaussian visible units. Its learning process is no longer unsupervised method, but guided by pairwise constraints.

3 PRELIMINARIES

In this section, the background of the pairwise constraints, RBM and Gaussian visible units is briefly summarized.

3.1 Pairwise Constraints

The priori knowledge of pairwise constraints is widely used in supervised and semi-supervised learning. There are two types of instance-level pairwise constraints: One is cannot-link constraints which instances should not be grouped together and the other is must-link constraints which instances should be grouped together. The must-link and cannot-link constraints define an instance-level relation of transitive binary. Consequently, two types of constraints may be derived from background knowledge about data set or labeled data. In this paper, we select labeled data from different groups randomly and ensure each group has the same ratio of labeled data to be selected. Then, the must-link constraints are produced by the selected same group labeled data and the cannot-link constraints are produced by the selected different group labeled data.

3.2 Restricted Boltzmann Machine

A RBM[2][3] is a two-layer network in which the first layer consists of visible units, and the second layer consists of hidden units. The symmetric undirected weights are used to connect the visible and hidden layers. There are no interior-layer connections with either the visible units or the hidden units. A classic RBM model is shown in Fig. 1. An energy function[56] of a joint configuration (\mathbf{v}, \mathbf{h}) between the visible layer and the hidden layer is given by:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

where $\mathbf{v} = (v_1, v_2, \dots, v_n)$ and $\mathbf{h} = (h_1, h_2, \dots, h_m)$ are the visible and the hidden vectors, a_i and b_j are their biases, n and m are the dimension of visible layer and hidden layer, respectively, w_{ij} is the connection weight matrix between the visible layer and the hidden layer. A probability distribution over vectors \mathbf{v} and \mathbf{h} is defined as

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2)$$

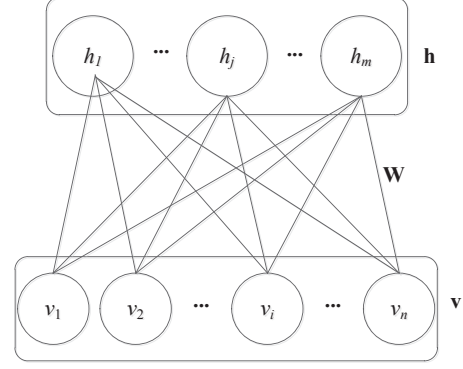


Fig. 1. Restricted Boltzmann Machine (RBM)

where Z is a “partition function” which is defined by summing over all possible pairs of hidden layer and visible layer:

$$Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \quad (3)$$

By means of summing over all the units of the hidden layer, the probability that the RBM assigns to the units of the visible layer \mathbf{v} is given by:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \quad (4)$$

The partial derivative of the log probability of Eq. (4) with respect to a weight is given by

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \quad (5)$$

where the angle brackets $\langle v_i h_j \rangle_{data}$ and $\langle v_i h_j \rangle_{model}$ are used to denote expectations of the distribution specified by the subscript *data* and *model*. In the log probability, a very simple learning rule for performing stochastic steepest ascent is given by:

$$\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \quad (6)$$

where ε is a learning rate.

It is easy to get $\langle v_i h_j \rangle_{data}$ because there is no direct connections among the hidden units. However, it is difficult to get unbiased sample of $\langle v_i h_j \rangle_{model}$. Hinton[3] proposed a faster learning algorithm with the CD learning and the change of learning parameter is given by:

$$\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}), \quad (7)$$

$$\Delta a_i = \varepsilon (\langle v_i \rangle_{data} - \langle v_i \rangle_{recon}), \quad (8)$$

$$\Delta b_j = \varepsilon (\langle h_j \rangle_{data} - \langle h_j \rangle_{recon}) \quad (9)$$

where $\langle v_i h_j \rangle_{recon}$ can be computed efficiently than $\langle v_i h_j \rangle_{model}$.

3.3 Gaussian Visible Units

Original RBMs were developed by binary stochastic units for the hidden and visible layers[3]. To deal with real-valued data such as natural images, one solution is that the binary visible units are replaced by linear units with independent Gaussian noise, but the hidden units remain binary, which is first suggested by[57]. The negative log probability is given by the following energy function:

$$-\log p(\mathbf{v}, \mathbf{h}) = E(\mathbf{v}, \mathbf{h}) = \sum_{i \in \text{visible}} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} \frac{v_i h_j w_{ij}}{\sigma_i} \quad (10)$$

where σ_i is the standard deviation of the Gaussian noise for visible unit i .

For each visible unit, it is easy to learn the variance of the noise, but it is difficult using \mathbf{CD}_1 because of taking long time[50][58]. Therefore, in many applications, it is easy to normalise the data to have unit variance and zero mean[50][59][60][61]. Then the reconstructed value of Gaussian visible units is equal to its input from the binary hidden units plus its bias.

4 PCGRBM MODEL AND ITS LEARNING ALGORITHM

We first propose a pairwise constraints restricted Boltzmann machine with Gaussian visible units(pcGRBM) model which the binary visible units are replaced by noise-free linear units and its learning procedure is guided by pairwise constraints. Then we give exact inference of the pcGRBM optimization. Finally, the corresponding learning algorithm is presented.

4.1 pcGRBM Model

Suppose that $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ is a p -dimensional original data set which has been normalized, $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ is a q -dimensional hidden code. The pairwise must-link constraints set of the reconstruction data is defined by $\mathcal{M} = \{(\mathbf{v}_s, \mathbf{v}_t) | \mathbf{v}_s, \mathbf{v}_t \text{ belongs to the same class}\}$ and pairwise cannot-link constraints set of the reconstruction data is given by $\mathcal{C} = \{(\mathbf{v}_s, \mathbf{v}_t) | \mathbf{v}_s, \mathbf{v}_t \text{ belongs to the different classes}\}$.

For training the parameters of the pcGRBM model, the first objective is that how to maximize the log probability of RBM with Gaussian visible units and the second objective is that how to maximize distance of all pairwise vectors which come from cannot-link set and minimize distance of all pairwise vectors which come from must-link set in reconstructed visible layer. Because of using noise-free reconstruction in the model, the reconstructed value of a Gaussian visible linear unit is equal to its input from the hidden units plus its bias. Then the objective function is given by

$$L(\theta, \mathcal{V}) = \frac{\lambda}{n} \sum_{\mathbf{v}_i \in \mathcal{V}} \log p(\mathbf{v}_i, \theta) + \left[\left(\frac{1-\lambda}{N_M} \sum_{\mathcal{M}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2 - \frac{1-\lambda}{N_C} \sum_{\mathcal{C}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2 \right) \right] \quad (11)$$

where $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ are the model parameters, $\lambda \in (0, 1)$ is a scale coefficient, N_M and N_C are the cardinality of the must-link pairwise constraints set \mathcal{M} and the cannot-link pairwise constraints set \mathcal{C} , respectively, $\frac{1}{n} \sum_{i=1}^n \log p(\mathbf{v}_i; \theta)$ is the average of the log-likelihood and $\|\cdot\|^2$ is the square of 2-norm.

The learning problem of the pcGRBM model is to get optimal or approximate optimal parameters θ , which minimize the objective function $L(\theta, \mathcal{V})$, i.e.,

$$\min \{L(\theta, \mathcal{V})\}. \quad (12)$$

4.2 pcGRBM Inference

For our first objective, we can use gradient descent to solve optimal problem, however, it is expensive to compute the gradient of the log probability. Recently, Karakida et al.[62] demonstrated that \mathbf{CD}_1 learning is simpler than ML learning in RBMs with Gaussian linear units. Then, we apply the \mathbf{CD}_1 learning method to obtain an approximation of the log probability gradient. For our second objective, we use the method of gradient descent to solve the optimization problem. The following main work is to compute the gradient of $\frac{1-\lambda}{N_M} \sum_{\mathcal{M}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2 - \frac{1-\lambda}{N_C} \sum_{\mathcal{C}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2$.

Firstly, we assume that

$$J_M(\mathbf{W}) = \frac{1}{N_M} \sum_{\mathcal{M}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2 \quad (13)$$

and

$$J_C(\mathbf{W}) = \frac{1}{N_C} \sum_{\mathcal{C}} \|\mathbf{h}_s \mathbf{W}^T - \mathbf{h}_t \mathbf{W}^T\|^2. \quad (14)$$

Then, the gradients of the $J_M(\mathbf{W})$ is

$$\frac{\partial J_M(\mathbf{W})}{\partial w_{ij}} = \frac{1}{N_M} \sum_{\mathcal{M}} \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T \frac{\partial \mathbf{W}(\mathbf{h}_s - \mathbf{h}_t)^T}{\partial w_{ij}} + \frac{\partial (\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T}{\partial w_{ij}} \mathbf{W}(\mathbf{h}_s - \mathbf{h}_t)^T \right] \quad (15)$$

and the gradients of the $J_C(\mathbf{W})$ is

$$\frac{\partial J_C(\mathbf{W})}{\partial w_{ij}} = \frac{1}{N_C} \sum_{\mathcal{C}} \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T \frac{\partial \mathbf{W}(\mathbf{h}_s - \mathbf{h}_t)^T}{\partial w_{ij}} + \frac{\partial (\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T}{\partial w_{ij}} \mathbf{W}(\mathbf{h}_s - \mathbf{h}_t)^T \right]. \quad (16)$$

In order to express concisely, we suppose that

$$\mathbf{h}' = (h_{s1} - h_{t1}, \dots, h_{sj} - h_{tj}, \dots, h_{sq} - h_{tq}) \quad (17)$$

where $h_{sk} - h_{tk} = 0, k \neq j, j = 1, 2, \dots, q$, and q is the dimension of the hidden layer.

Then, the gradient of the $J_M(\mathbf{W})$ takes the form

$$\frac{\partial J_M(\mathbf{W})}{\partial w_{ij}} = \frac{1}{N_M} \sum_{\mathcal{M}} \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W}(\mathbf{h}_s - \mathbf{h}_t)^T \right]. \quad (18)$$

In like manner, the gradient of the $J_C(\mathbf{W})$ takes the form

$$\frac{\partial J_C(\mathbf{W})}{w_{ij}} = \frac{1}{N_C} \sum_c \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W} (\mathbf{h}_s - \mathbf{h}_t)^T \right]. \quad (19)$$

So, the gradient of the objective function is as follows.

$$\begin{aligned} \nabla w_{ij} = & \lambda \varepsilon (< v_i h_j >_{data} - < v_i h_j >_{recon}) + \\ & \frac{1}{N_M} \sum_{\mathcal{M}} \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W} (\mathbf{h}_s - \mathbf{h}_t)^T \right] - \\ & \frac{1}{N_C} \sum_c \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W} (\mathbf{h}_s - \mathbf{h}_t)^T \right]. \end{aligned} \quad (20)$$

It is obvious that $\frac{\partial J_M(\mathbf{W})}{a_i} = 0$, $\frac{\partial J_C(\mathbf{W})}{a_i} = 0$, $\frac{\partial J_M(\mathbf{W})}{b_j} = 0$ and $\frac{\partial J_C(\mathbf{W})}{b_j} = 0$. So, in the pcGRBM model, we use Eq. (8) and Eq. (9) to update the biases a_i and b_j .

Finally, the updating rulers of connection weights \mathbf{W} of the pcGRBM model takes the form

$$\begin{aligned} w_{ij}^{(\tau+1)} = & w_{ij}^{(\tau)} + \lambda \varepsilon (< v_i h_j >_{data} - < v_i h_j >_{recon}) + \\ & \frac{1}{N_M} \sum_{\mathcal{M}} \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W} (\mathbf{h}_s - \mathbf{h}_t)^T \right] - \\ & \frac{1}{N_C} \sum_c \left[(\mathbf{h}_s - \mathbf{h}_t) \mathbf{W}^T (\mathbf{h}')^T + \mathbf{h}' \mathbf{W} (\mathbf{h}_s - \mathbf{h}_t)^T \right]. \end{aligned} \quad (21)$$

4.3 pcGRBM Learning Algorithm

According to the above inference, the learning algorithm for pcGRBM is summarized as follows.

Algorithm 1 Learning for pcGRBM

Input: ε is the learning rate;

\mathcal{V} is a p -dimensional data set;

λ is a scale coefficient;

N_M is the cardinality of the must-link pairwise constraints set;

N_C is the cardinality of the cannot-link pairwise constraints set;

\mathcal{M} is must-link pairwise constraints set;

\mathcal{C} is must-link pairwise constraints set.

Output: $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$, \mathbf{W} is connection weights, \mathbf{a} is visible biases, \mathbf{b} is hidden biases.

Initializing ε , λ , N_M , N_C , \mathbf{W} , \mathbf{a} , \mathbf{b} ;

For each iteration **do**

For all hidden units j **do**

compute $p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij})$;

sample $h_j \in \{0, 1\}$ from $p(h_j = 1|\mathbf{v})$;

End For

For all visible units i **do**

compute reconstructed value $v_i = a_i + \sum_j h_j w_{ij}$;

End For

compute the gradient of the $\frac{\partial J_M(\mathbf{W})}{w_{ij}}$ by using Eq. (18);

compute the gradient of the $\frac{\partial J_C(\mathbf{W})}{w_{ij}}$ by using Eq. (19);

TABLE 1
Summary of the data sets.

No.	Dataset	classes	Instances	features
1	alphabet	3	814	892
2	ambulances	3	930	892
3	bed	3	888	892
4	beret	3	876	892
5	beverage	3	873	892
6	bike	3	839	892
7	billiard	3	859	892
8	blog	3	943	892
9	blood	3	866	892
10	bonsai	3	867	892
11	book	3	896	892
12	bread	3	885	892
13	breakfast	3	895	892
14	building	3	911	892
15	vegetable	3	872	892
16	virus	3	871	892

update connection weights \mathbf{W} matrix by using Eq. (21);

update visible biases \mathbf{a} matrix by using Eq. (8);

update hidden biases \mathbf{b} matrix by using Eq. (9);

End For

return \mathbf{W} , \mathbf{a} , \mathbf{b} .

5 RESULTS AND DISCUSSION

In this section, we introduce the datasets, define the experimental setup, and discuss about experimental results.

5.1 DataSets

We used the Microsoft Research Asia Multimedia (MSRA-MM)[63] which contains two sub-datasets, e.g., a video dataset and an image dataset. The image part contains 1,011,738 images and the video part contains 23,517 videos. To evaluate our pcGRBM model, we use 16 image datasets (alphabet, ambulances, bed, beret, beverage, bike, billiard, blog, blood, bonsai, book, bread, breakfast, building, vegetable and virus) from image part for our experiments. The summary of the datasets are listed in Table I.

5.2 Experimental Setup

The goal of the experiments is to study the following aspects:

- Dose the pairwise constraints guide the encoding procedure of traditional RBM?
- How dose unsupervised clustering algorithms based on pcGRBM model compare with their semi-supervised clustering algorithms?
- How dose unsupervised clustering algorithms based on pcGRBM model compare with these algorithms based on traditional RBM?

To verify the features of pcGRBM contain guiding information whether or not, we use the output of pcGRBM as input of unsupervised clustering algorithm. In our experiments, we choose K-means, affinity propagation (AP)[64], SP clustering algorithms as examples. Then, we present three algorithms

TABLE 2

The average accuracies and variance results of K-means, SP, AP, Cop-Kmeans, Semi-SP, Semi-AP, Kmeans.grbm, SP.grbm, AP.grbm, Kmeans.pcgrbm, SP.pcgrbm, AP.pcgrbm algorithms.

Dataset	K-means	SP	AP	Cop-Kmeans	Semi-SP	Semi-AP	Kmeans.grbm	SP.grbm	AP.grbm	Kmeans.pcgrbm	SP.pcgrbm	AP.pcgrbm
alphabet	0.4144±0.0066	0.4171±0.0004	0.4042±0.0000	0.4047±0.0385	0.3748±0.0230	0.4221±0.1144	0.4102±0.0360	0.3940±0.1413	0.4290±0.0935	0.4195±0.0503	0.4300±0.0375	0.4303±0.0415
ambulances	0.4397±0.3867	0.4157±0.0045	0.3968±0.0000	0.4340±0.2465	0.4166±0.6353	0.4116±0.0518	0.4477±0.0739	0.4741±0.9110	0.4222±0.1193	0.4673±0.9257	0.4663±0.5724	0.4788±0.9363
bed	0.4694±0.0043	0.4152±0.0018	0.4144±0.0000	0.4252±0.1781	0.4282±0.7285	0.4226±0.1091	0.4652±0.1671	0.4375±0.4247	0.4456±0.3005	0.4840±0.5696	0.4816±0.5283	0.4843±0.5291
beret	0.4331±0.1144	0.3818±0.0115	0.3995±0.0000	0.4510±0.2633	0.4330±0.5226	0.4277±0.1478	0.4244±0.2909	0.4586±1.2623	0.4369±0.5376	0.4866±0.4695	0.4838±0.5934	0.4752±0.2543
beverage	0.4226±0.0080	0.3998±0.0193	0.4840±0.0000	0.4523±0.1751	0.3911±0.1237	0.4874±0.1003	0.4481±0.1513	0.4252±0.2096	0.4418±0.2565	0.4700±0.3233	0.4638±0.3584	0.4688±0.2848
bike	0.4139±0.0194	0.3661±0.0005	0.4017±0.0000	0.4088±0.1548	0.3681±0.0337	0.3956±0.0185	0.4089±0.0241	0.3874±0.2335	0.3883±0.0701	0.4237±0.2545	0.4094±0.2004	0.4285±0.2391
billiard	0.4477±0.0681	0.3819±0.0013	0.3818±0.0000	0.4334±0.1746	0.3722±0.0928	0.3960±0.0362	0.4288±0.1680	0.4181±0.1053	0.4250±0.2380	0.4518±0.1644	0.4688±0.2290	0.4632±0.1727
blog	0.4189±0.0664	0.3625±0.0006	0.3860±0.0000	0.4093±0.2979	0.4370±0.2778	0.3852±0.0311	0.4351±0.0252	0.5001±0.6142	0.4142±0.0721	0.5090±0.1194	0.4685±0.2493	0.4885±0.1608
blood	0.4670±0.0242	0.3698±0.0015	0.5589±0.0000	0.4533±0.0905	0.3882±0.0463	0.4889±0.1963	0.4585±0.0714	0.5520±2.4200	0.4851±0.1072	0.5409±0.9060	0.5555±0.5221	0.5592±0.4185
bonsai	0.4246±0.0212	0.3743±0.0004	0.3783±0.0000	0.4444±0.2094	0.4322±0.8488	0.4027±0.0576	0.4253±0.0419	0.4478±0.8064	0.4376±0.2303	0.5063±0.4142	0.4836±0.9142	0.4844±0.8520
book	0.3776±0.0071	0.3695±0.0001	0.4230±0.0000	0.3904±0.0363	0.3994±0.1333	0.4131±0.0422	0.4016±0.0482	0.4029±0.5629	0.3962±0.0381	0.4403±0.2994	0.4334±0.1442	0.4386±0.3080
bread	0.4358±0.0308	0.4149±0.0004	0.4384±0.0000	0.4540±0.2315	0.4068±0.1296	0.4395±0.0459	0.4401±0.0977	0.4208±0.7148	0.4328±0.1530	0.4480±0.6221	0.4642±0.4292	0.4463±0.03739
breakfast	0.5014±0.4058	0.5120±0.1828	0.4123±0.0000	0.4823±0.2297	0.4091±0.1916	0.4384±0.1474	0.4597±0.2587	0.3785±0.0594	0.4640±0.1685	0.5095±0.8512	0.4884±0.6350	0.5036±0.8468
building	0.5203±0.1972	0.4499±0.0058	0.4874±0.0000	0.5386±0.4887	0.4020±0.3751	0.4633±0.1135	0.4419±0.0639	0.5411±1.0742	0.4409±0.2259	0.5516±0.8652	0.5512±0.7238	0.5541±0.4670
vegetable	0.4157±0.0450	0.3884±0.0004	0.4243±0.0000	0.4263±0.1187	0.4185±0.2175	0.4287±0.0155	0.4188±0.0427	0.4274±0.2127	0.4774±0.0725	0.4823±0.3213	0.4719±0.4277	0.4069±0.0242
virus	0.4036±0.0292	0.3759±0.0066	0.3823±0.0000	0.4054±0.0524	0.3648±0.0231	0.3877±0.0069	0.3987±0.0090	0.3537±0.0644	0.4057±0.0321	0.4103±0.0304	0.4055±0.0256	0.4069±0.0242
Average	0.4378	0.3997	0.4231	0.4385	0.4026	0.4253	0.4321	0.4387	0.4311	0.4748	0.4713	0.4739

TABLE 3

Aligned observations of K-means, SP, AP, Cop-Kmeans, Semi-SP, Semi-AP, Kmeans.grbm, SP.grbm, AP.grbm, Kmeans.pcgrbm, SP.pcgrbm, AP.pcgrbm algorithms selected in the experimental study. The ranks in the parentheses are used in the computation of the Friedman Aligned Ranks test. The smaller the better.

Dataset	K-means	SP	AP	Cop-Kmeans	Semi-SP	Semi-AP	Kmeans.grbm	SP.grbm	AP.grbm	Kmeans.pcgrbm	SP.pcgrbm	AP.pcgrbm	Total
alphabet	0.0016(91)	0.0044(84)	-0.0086(122)	-0.0054(113)	-0.0379(168)	0.0094(74)	-0.0026(101)	-0.0188(141)	0.0162(60)	0.0068(80)	0.0172(58)	0.0176(56)	1148
ambulances	0.0004(96)	-0.0235(149)	-0.0425(173)	-0.0053(112)	-0.0227(147)	-0.0276(156)	0.0085(77)	0.0349(32)	-0.0171(138)	0.0281(41)	0.0271(43)	0.0396(27)	1191
bed	0.0216(52)	-0.0325(162)	-0.0334(163)	-0.0226(146)	-0.0196(142)	-0.0251(152)	0.0174(57)	-0.0103(125)	-0.0022(100)	0.0362(31)	0.0339(36)	0.0366(30)	1196
beret	-0.0075(117)	-0.0587(186)	-0.0410(172)	0.0105(71)	-0.0076(118)	-0.0178(139)	-0.0161(136)	0.0180(55)	-0.0037(103)	0.0461(19)	0.0432(22)	0.0347(33)	1171
beverage	-0.0238(150)	-0.0465(176)	0.0341(34)	0.0060(82)	-0.0553(184)	0.0411(24)	-0.0018(89)	-0.0210(144)	-0.0045(109)	0.0237(49)	0.0219(51)	0.0225(50)	1142
bike	0.0139(64)	-0.0340(164)	0.0016(90)	0.0088(76)	-0.0320(161)	-0.0044(108)	0.0089(75)	-0.0126(130)	-0.0117(128)	0.0237(48)	0.0094(73)	0.0284(40)	1157
billiard	0.0253(45)	-0.0405(170)	-0.0406(171)	0.0110(70)	-0.0502(179)	-0.0264(154)	0.0064(81)	-0.0043(107)	0.0026(87)	0.0294(39)	0.0464(17)	0.0408(25)	1145
blog	-0.0156(132)	-0.0720(188)	-0.0485(177)	-0.0252(153)	0.0025(88)	-0.0494(178)	0.0005(95)	0.0655(6)	-0.0203(143)	0.0745(1)	0.0340(35)	0.0540(11)	1207
blood	-0.0228(148)	-0.1199(192)	0.0691(4)	-0.0364(166)	-0.1016(191)	-0.0008(97)	-0.0312(159)	0.0622(7)	-0.0047(111)	0.0511(12)	0.0658(5)	0.0694(3)	1095
bonsai	-0.0122(129)	-0.0624(187)	-0.0585(185)	0.0076(78)	-0.0046(110)	-0.0341(165)	-0.0115(127)	0.0110(69)	0.0008(93)	0.0695(2)	0.0468(16)	0.0476(15)	1176
book	-0.0296(157)	-0.0377(167)	0.0158(61)	-0.0168(137)	-0.0077(119)	0.0059(83)	-0.0056(114)	-0.0043(106)	-0.0110(126)	0.0331(37)	0.0262(44)	0.0315(38)	1189
bread	-0.0009(98)	0.0219(145)	0.0016(92)	0.0172(59)	-0.0300(158)	0.0027(86)	0.0033(85)	-0.0160(135)	-0.0040(105)	0.0112(68)	0.0274(42)	0.0095(72)	1145
breakfast	0.0381(28)	0.0487(13)	-0.0510(180)	0.0191(53)	-0.0542(182)	-0.0248(151)	-0.0036(102)	-0.0848(189)	0.0007(94)	0.0462(18)	0.0251(46)	0.0403(26)	1082
building	0.0251(47)	-0.0453(174)	-0.0078(120)	0.0434(21)	-0.0932(190)	-0.0319(160)	-0.0533(181)	0.0459(20)	-0.0543(183)	0.0564(9)	0.0560(10)	0.0589(8)	1123
vegetable	-0.0187(140)	-0.0460(175)	-0.0101(124)	-0.0081(121)	-0.0159(134)	-0.0057(115)	-0.0156(131)	-0.0070(116)	-0.0015(99)	0.0430(23)	0.0480(14)	0.0375(29)	1221
virus	0.0119(67)	-0.0158(133)	-0.0094(123)	0.0137(66)	-0.0270(155)	-0.0040(104)	0.0007(79)	-0.0380(169)	0.0140(63)	0.0186(54)	0.0138(65)	0.0152(62)	1140
Total	1561	2465	1991	1524	2426	1946	1689	1551	1742	531	577	525	18528
Average rank	97.5625	154.0625	124.4375	95.2500	151.6250	121.6250	105.5625	96.9375	108.8750	33.1875	36.0625	32.8125	

TABLE 4

Aligned observations of K-means, SP, AP, Cop-Kmeans, Semi-SP, Semi-AP, Kmeans.grbm, SP.grbm, AP.grbm, Kmeans.pcgrbm, SP.pcgrbm, AP.pcgrbm algorithms selected in the experimental study. The purity is used to evaluate performance of these 12 clustering algorithms. The larger the better.

Dataset	K-means	SP	AP	Cop-Kmeans	Semi-SP	Semi-AP	Kmeans.grbm	SP.grbm	AP.grbm	Kmeans.pcgrbm	SP.pcgrbm	AP.pcgrbm
alphabet	0.8394	0.8445	0.8453	0.8457	0.8597	0.8396	0.8530	0.8637	0.8516	0.8600	0.8584	0.8604
ambulances	0.7479	0.7517	0.7525	0.7475	0.7463	0.7476	0.7540	0.7511	0.7570	0.7750	0.7740	0.7754
bed	0.7482	0.7502	0.7508	0.7689	0.7654	0.7504	0.7728	0.7691	0.7770	0.7895	0.7907	0.7911
beret	0.7034	0.7062	0.7095	0.7039	0.7011	0.7074	0.7136	0.7109	0.7155	0.7294	0.7291	0.7276
bike	0.7669	0.7665	0.7701	0.7760	0.7818	0.7661	0.7888	0.7897	0.7872	0.7983	0.7973	0.7955
billiard	0.9193	0.9213	0.9211	0.9209	0.9045	0.9200	0.9214	0.9126	0.9215	0.9354	0.9364	0.9362
bike	0.9082	0.9124	0.9293	0.9147	0.9204	0.9219	0.9270	0.9270	0.9285	0.9329	0.9327	0.9320
billiard	0.6728	0.6744	0.6684	0.6744	0.6591	0.6704	0.6734	0.6688	0.6729	0.6916	0.6922	0.6921
blog	0.4590	0.4612	0.4608	0.4592	0.4505	0.4575	0.4563	0.4571	0.4568	0.4774	0.4781	0.4779
blood	0.7534	0.7544	0.7479	0.7546	0.7363	0.7466	0.7513	0.7448	0.7515	0.7655	0.7662	0.7656
bonsai	0.8665	0.8666	0.8731	0.8740	0.8602	0.8706	0.8756	0.8684	0.8755	0.8845	0.8841	0.8853
book	0.7343	0.7355	0.7299	0.7351	0.7515	0.7350	0.7494	0.7572	0.7521	0.7756	0.7756	0.7764
breakfast	0.7770	0.7762	0.8051	0.7989	0.8411	0.8095	0.8099	0.8434	0.8168	0.8567	0.8618	0.8569
building	0.6396	0.6439	0.6637	0.6328	0.6807	0.6548	0.6769	0.6884	0.6728	0.7019	0.6986	0.7018
vegetable	0.8331	0.8272	0.8457	0.8327	0.8407	0.8466	0.8466	0.8486	0.8439	0.8577	0.8581	0.8578
virus	0.9559	0.9612	0.9619	0.9483	0.9620	0.9605	0.9618	0.9632	0.9581	0.9849	0.9854	0.9863
Average	0.7703	0.7721	0.7772	0.7742	0.7788	0.7753	0.7831	0.7853	0.7837	0.8010	0.8012	0.8011

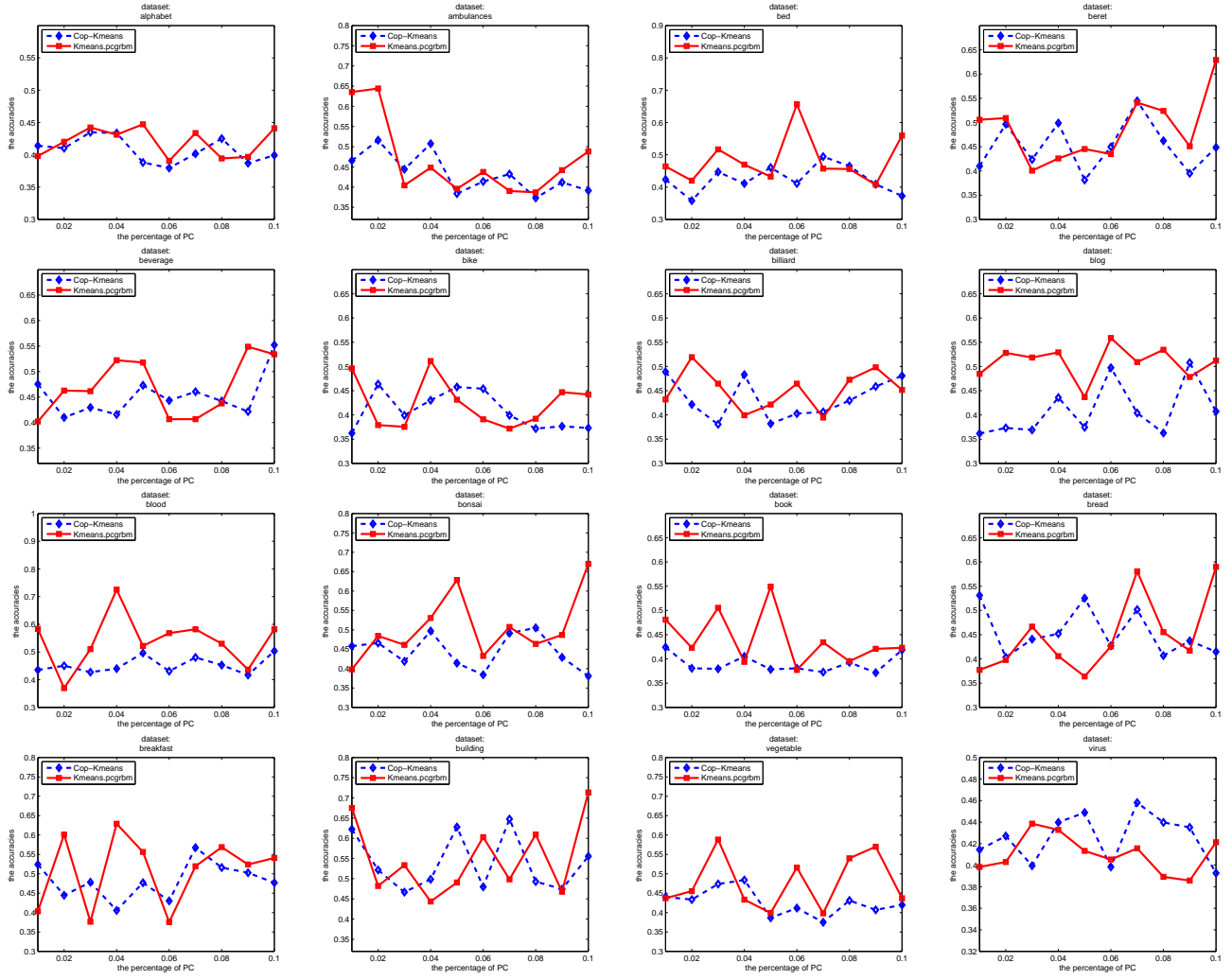


Fig. 2. Cop-Kmeans and Kmeans.pcgrbm results on alphabet, ambulances, bed, beret, beverage, bike, billiard, blog, blood, bonsai, book, bread, breakfast, building, vegetable and virus data sets with increasing percentage of pairwise constraints (PC) from 1% to 10% in steps of 1%.

which based on pcGRBM model for clustering task, termed as Kmeans.pcgrbm, AP.pcgrbm and SP.pcgrbm, their structure are shown in Fig.5. Similarly, we also present three algorithms which based on traditional RBM with Gaussian visible units for clustering task, called as Kmeans.grbm, AP.grbm and SP.grbm. In fact, Kmeans.pcgrbm, AP.pcgrbm and SP.pcgrbm are semi-supervised clustering algorithms with instance-level guiding of pairwise constraints, but Kmeans.grbm, AP.grbm and SP.grbm are unsupervised methods.

Firstly, we compare the clustering performance of the proposed algorithms (Kmeans.pcgrbm, AP.pcgrbm and SP.pcgrbm) with original K-means, AP and SP clustering algorithms, respectively. Secondly, the proposed algorithms are used to compare with Cop-Kmeans[42], Semi-SP[43] and Semi-AP[44], respectively. Finally, we use unsupervised algorithms that are Kmeans.grbm, AP.grbm and SP.grbm to compare with the proposed algorithms.

To evaluate the performance of the clustering algorithms, we adopt three widely used metrics: clustering accuracy[65], clustering purity[66] and clustering rank[67] as the evaluation measure.

5.3 Results

5.3.1 The pcGRBM for Clustering VS Unsupervised Algorithms

In this section, we compare unsupervised clustering of K-means, SP and AP with Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm which based on the pcGRBM by evaluation of average accuracy, average rank and average purity. From Table II, the average accuracies of K-means, SP and AP algorithms are 43.78%, 39.97% and 42.31%, respectively, but the average accuracies of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms raise to 47.48%, 47.13% and 47.39%, respectively. The average ranks of K-means, SP and AP algorithms are shown in Table III, their values are 97.5625, 154.0625 and 124.4375, respectively, but the average ranks of

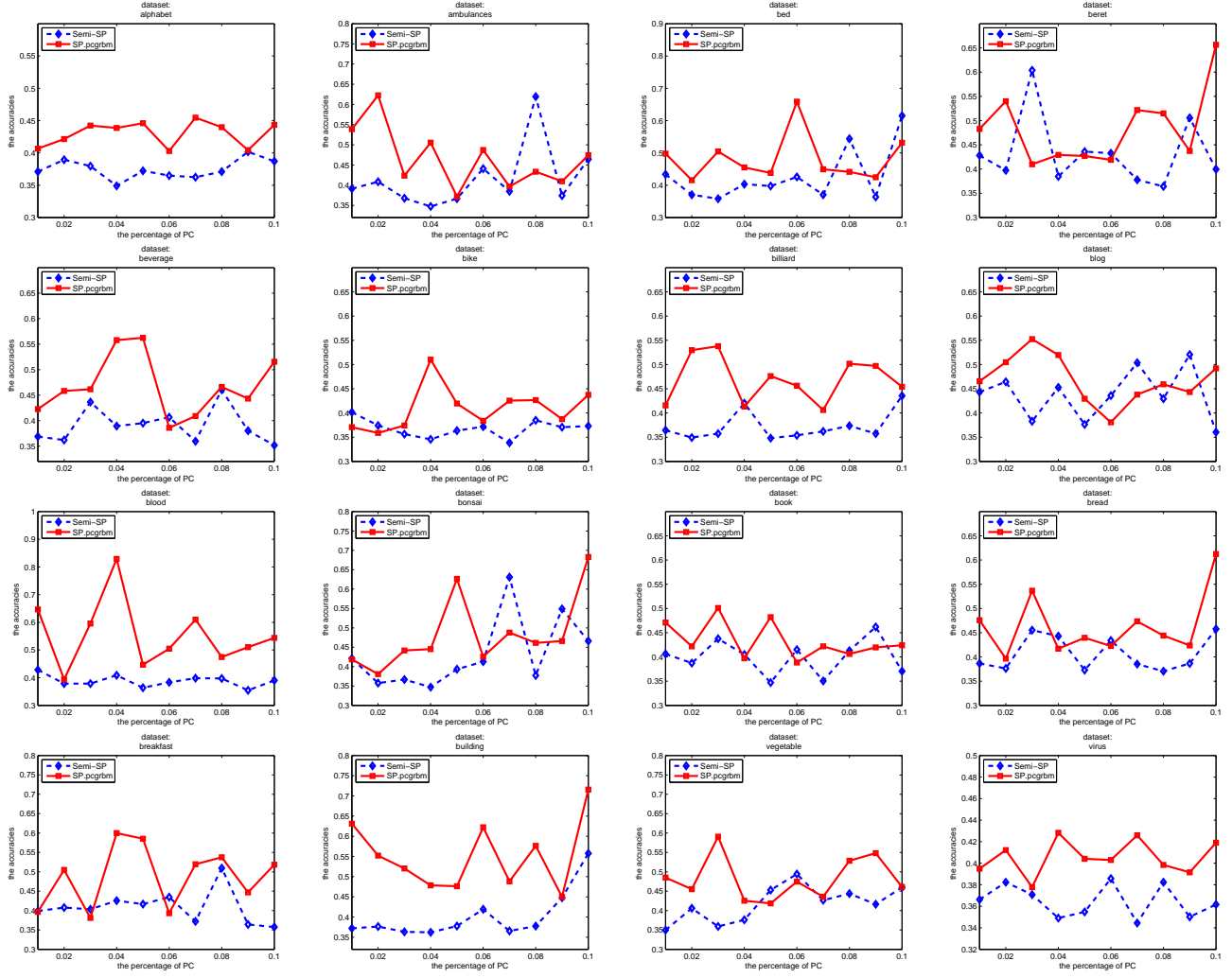


Fig. 3. Semi-SP and SP.pcgrbm results on alphabet, ambulances, bed, beret, beverage, bike, billiard, blog, blood, bonsai, book, bread, breakfast, building, vegetable and virus data sets with increasing percentage of pairwise constraints (PC) from 1% to 10% in steps of 1%.

Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms reduce to 33.1875, 36.0625 and 32.8125, respectively. The smaller the rank value means the better the algorithm. From Table IV, the average purities of K-means, SP and AP algorithms are 0.7703, 0.7721 and 0.7772, respectively, but the average purities of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms raise to 0.8010, 0.8012 and 0.8011, respectively. A greater purity indicates a better algorithm. From all above results, it is obvious that clustering by the pcGRBM is better than the original unsupervised clustering.

From the last three columns of Table II, there are more variance volatility of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm than those of other algorithms because of the effect of pairwise constraints.

5.3.2 The pcGRBM for Clustering VS Semi-supervised Algorithms

In this section, we make further comparison among semi-supervised clustering of Cop-kmeans, Semi-SP and Semi-AP with Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm by evalu-

ation of average accuracy, average rank and average purity. In addition, the comparison of average accuracy is shown in Figs .2-4, respectively. From Table II, the average accuracies of Cop-kmeans, Semi-SP and Semi-AP with Kmeans.pcgrbm algorithms are 43.85%, 40.26% and 42.53%, respectively. The pcGRBM raise the average accuracies by 3.98%, 6.87% and 5.09%, respectively. From Table III, the average ranks of Cop-kmeans, Semi-SP and Semi-AP algorithms are 95.2500, 151.6250 and 121.6250, respectively, however, the average ranks of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms are 33.1875, 36.0625 and 32.8125, respectively. The smaller the rank value means the better the algorithm. The average purities of Cop-kmeans, Semi-SP and Semi-AP algorithms are shown in Table IV, their values are 0.7742, 0.7788 and 0.7753, respectively. From all above results, it is obvious that the pcGRBM for clustering is better than the semi-supervised clustering.

We plot the experiment results with the increasing percentage of pairwise constraints which ranges from 1% to 10% in steps of 1% for Cop-Kmeans and Kmeans.pcgrbm

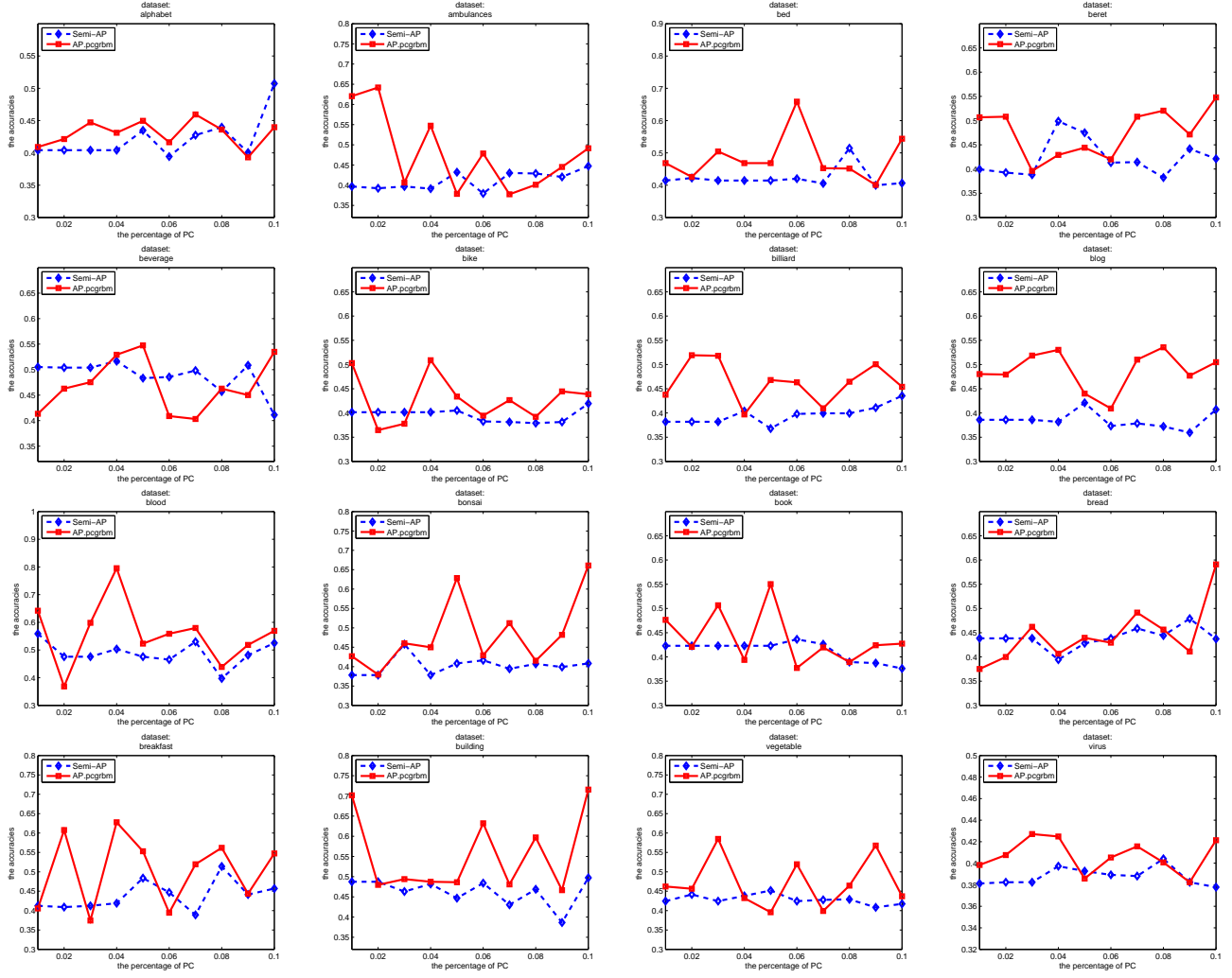


Fig. 4. Semi-AP and AP.pcgrbm results with increasing percentage of pairwise constraints (PC) from 1% to 10% in steps of 1%.

in Fig .2, Semi-SP and SP.pcgrbm in Fig .3, Semi-AP and AP.pcgrbm in Fig .4. From Figs .2-4, we can see that the accuracy of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm can not maintain complete synchronous increases as the percentage of pairwise constraints, however, the average accuracies of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm are higher than Cop-Kmeans, Semi-SP and Semi-AP, respectively.

5.3.3 The pcGRBM VS RBM with Gaussian Visible Units for Clustering

The pcGRBM and RBM with Gaussian visible have ability to extract features, but, which one shows better performance for clustering task? In order to compare the representation capability between the pcGRBM and RBM without any guiding of pairwise constraints, we design a structure of clustering algorithm in which the features of RBM with Gaussian visible units is used as input of unsupervised clustering. In our experiment, we use three clustering algorithms base on this structure which are termed as Kmeans.grbm, SP.grbm and AP.grbm algorithms to compare to Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm. From Table II,

the average accuracies of kmeans.grbm, SP.grbm and AP.grbm algorithms are 43.321%, 43.387% and 43.11%, respectively, however, Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms raise the average accuracies by 4.27%, 3.26% and 4.28%, respectively. The average ranks of Kmeans.grbm, SP.grbm and AP.grbm algorithms are shown in Table III. The results are 105.5625, 96.9375 and 108.8750, respectively, however the average ranks of Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms reduce to 102.375, 60.875 and 76.0625, respectively. Table IV shows the average purities of kmeans.grbm, SP.grbm and AP.grbm. The values are 0.7831, 0.7853 and 0.7837, respectively. From all above results, it is obvious that the pcGRBM is better than RBM for clustering.

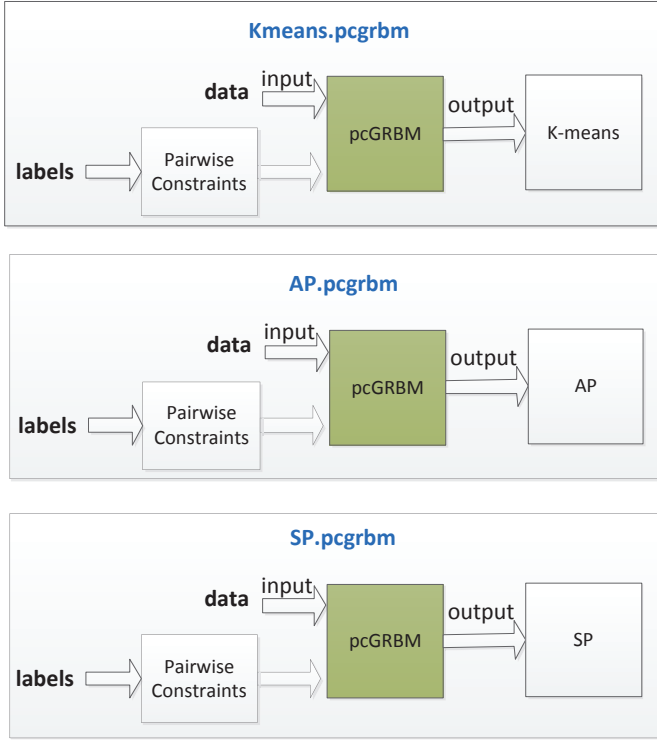


Fig. 5. Three structures of clustering algorithm base on pcGRBM model

5.3.4 The Rank

We compare twelve algorithms and sixteen data sets by means of the Aligned Friedman test statistic[67] which is given by

$$T = \frac{(G-1) \left[\sum_{j=1}^G \hat{R}_{.j}^2 - (GD^2/4)(GD+1)^2 \right]}{\{[GD(GD+1)(2GD+1)]/6\} - (1/G) \sum_{i=1}^D \hat{R}_i^2} \quad (22)$$

where \hat{R}_i is the rank sum of the j th algorithm, $\hat{R}_{.j}$ is the rank sum of the i th data set, D is the number of data set and G is the number of algorithm.

In our experiments, all pairwise constraints come from labels information. We choose 1% to 10% pairwise constraints in steps of 1%. The average rank value is smaller the algorithm is better. As we can see from Table III, the average rank of K-means, SP, AP, Cop-K-means, Semi-SP, Semi-AP, Kmeans.grbm, SP.grbm, AP.grbm, Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms are 97.5625, 154.0625, 124.4375, 95.2500, 151.6250, 121.6250, 105.5625, 96.9375, 108.8750, 33.1875, 36.0625 and 32.8125, respectively. It is easy to know that the least average rank is AP.pcgrbm algorithm with a value of 32.8125. From results on Table III, we can see that Kmeans.pcgrbm, SP.pcgrbm, AP.pcgrbm algorithms which based on pcGRBM are better than other nine algorithms. We check whether the measured sum of the ranks is significantly different from the average value of the total

ranks $\hat{R}_j = 1544$ expected under the null hypothesis:

$$\begin{aligned} \sum_{j=1}^k \hat{R}_{.j}^2 &= 1561^2 + 2465^2 + 1991^2 + 1524^2 + 2426^2 + 1946^2 \\ &\quad + 1689^2 + 1551^2 + 1742^2 + 531^2 + 577^2 + 525^2 \\ &= 33655396, \end{aligned} \quad (23)$$

$$\begin{aligned} \sum_{j=1}^k \hat{R}_{i..}^2 &= 1148^2 + 1191^2 + 1196^2 + 1171^2 + 1142^2 + 1157^2 \\ &\quad + 1145^2 + 1207^2 + 1095^2 + 1176^2 + 1189^2 + 1145^2 \\ &\quad + 1082^2 + 1123^2 + 1221^2 + 1140^2 = 21477770, \end{aligned} \quad (24)$$

$T =$

$$\begin{aligned} &\frac{\{(12-1)(33655396 - (12 \times 16^2/4)(12 \times 16 + 1)^2)\}}{(12 \times 16(12 \times 16 + 1)(2 \times 12 \times 16 + 1))/12 - 21477770/12} \\ &= 54.9855. \end{aligned} \quad (25)$$

T is the chi-square distribution with 11 degrees of freedom because we use nine algorithms and sixteen data sets. For one tailed test, the p -value is 0.00000001 which is computed by $\chi^2(11)$ distribution and the p -value is 0.000000001 for two-tailed test. Then, the null hypothesis is rejected at high level significance. The experimental results of algorithms are significantly different because the p -values are far less than 0.05.

6 CONCLUSION

In this paper, we proposed a novel pcGRBM model, the learning procedure of which is guided by the pairwise constraints and the process of encoding is conducted under guidance. Then, some pairwise hidden features of pcGRBM flock together and another part of them are separated by the guidances. In the process of learning pcGRBM, CD learning is used to approximate ML learning and pairwise constraints are iterated transitions between visible and hidden units. Then, the background of pairwise constraints are encoded in hidden layer features of pcGRBM. In order to testify the availability of pcGRBM, the features of the hidden layer of the pcGRBM are used as input 'data' for clustering tasks. The experimental results showed that the performance of the Kmeans.pcgrbm, SP.pcgrbm and AP.pcgrbm algorithms which based on pcGRBM for clustering tasks are better than their classic unsupervised clustering algorithms (K-means, SP, AP), semi-supervised clustering algorithms (Cop-kmeans, Semi-SP, Semi-AP) and even better than Kmeans.grbm, SP.grbm and AP.grbm which based on RBM with Gaussian visible units without guiding of pairwise constraints.

There are several interesting questions in our future studies. For example, how to design deep networks based on the pcGRBM. How to strengthen pairwise constraints information when the layer of the deep network becomes deeper and deeper. How many dimensions in hidden layer can enhance the performance for clustering.

7 ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation of China (No. 61573292).

REFERENCES

- [1] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Citeseer, 1983, pp. 448–453.
- [2] G. Hinton and T. Sejnowski, "Learning and relearning in boltzmann machines," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, pp. 282–317, 1986.
- [3] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [4] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. Citeseer, 2005, pp. 33–40.
- [5] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [6] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, vol. 19, p. 153, 2007.
- [7] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep boltzmann machines," *Neural Computation*, vol. 24, no. 8, pp. 1967–2006, 2012.
- [8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from over-fitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [10] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.
- [11] O. Fink, E. Zio, and U. Weidmann, "Fuzzy classification with restricted boltzman machines and echo-state networks for predicting potential railway door system failures," *Reliability, IEEE Transactions on*, vol. 64, no. 3, pp. 861–868, 2015.
- [12] Y. Chen, X. Zhao, and X. Jia, "Spectral-spatial classification of hyperspectral data based on deep belief network," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 8, no. 6, pp. 2381–2392, 2015.
- [13] S. Elfving, E. Uchibe, and K. Doya, "Expected energy-based restricted boltzmann machine for classification," *Neural Networks*, vol. 64, pp. 29–38, 2015.
- [14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [15] Y. W. Teh and G. E. Hinton, "Rate-coded restricted boltzmann machines for face recognition," *Advances in Neural Information Processing Systems*, pp. 908–914, 2001.
- [16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
- [17] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in Neural Information Processing Systems*, 2009, pp. 1607–1614.
- [18] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.
- [19] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 4, pp. 778–784, 2014.
- [20] S. Nie, Z. Wang, and Q. Ji, "A generative restricted boltzmann machine based method for high-dimensional motion data modeling," *Computer Vision and Image Understanding*, vol. 136(C), pp. 14–22, 2015.
- [21] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [22] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using rnn pre-trained by recurrent temporal restricted boltzmann machines," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 3, pp. 580–587, 2015.
- [23] X. Yang, Q. Chen, S. Zhou, and X. Wang, "Deep belief networks for automatic music genre classification," in *Twelfth Annual Conference of the International Speech Communication Association*, vol. 8, no. 11, 2011, pp. 13–16.
- [24] M. Yuan, H. Tang, and H. Li, "Real-time keypoint recognition using restricted boltzmann machine," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 11, pp. 2119–2126, 2014.
- [25] L. Nie, A. Kumar, and S. Zhan, "Periocular recognition using unsupervised convolutional rbm feature learning," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 399–404.
- [26] C. Chen, C.-Y. Zhang, L. Chen, and M. Gan, "Fuzzy restricted boltzmann machine for the enhancement of deep learning," *Fuzzy Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 2163–2173, 2015.
- [27] Q. Yu, Y. Hou, X. Zhao, and G. Cheng, "Rényi divergence based generalization for learning of classification restricted boltzmann machines," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 692–697.
- [28] A. Courville, G. Desjardins, J. Bergstra, and Y. Bengio, "The spike-and-slab rbm and extensions to discrete and sparse data distributions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 9, pp. 1874–1887, 2014.
- [29] N. Wang, J. Melchior, and L. Wiskott, "Gaussian-binary restricted boltzmann machines on modeling natural image statistics," *arXiv preprint arXiv:1401.5900*, 2014.
- [30] C. Ekanadham, S. Reader, and H. Lee, "Sparse deep belief net models for visual area v2," *Advances in Neural Information Processing Systems*, vol. 20, 2007.
- [31] N. Srivastava, R. R. Salakhutdinov, and G. E. Hinton, "Modeling documents with deep boltzmann machines," *arXiv preprint arXiv:1309.6865*, 2013.
- [32] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine," in *Advances in Neural Information Processing Systems*, 2009, pp. 1601–1608.
- [33] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li, "Unsupervised 3d local feature learning by circle convolutional restricted boltzmann machine," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5331–5344, 2016.
- [34] T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2032–2045, 2016.
- [35] F. Zhao, Y. Huang, L. Wang, T. Xiang, and T. Tan, "Learning relevance restricted boltzmann machine for unstructured group activity and event understanding," *International Journal of Computer Vision*, pp. 1–17, 2016.
- [36] M. V. Giuffrida and S. A. Tsafaris, "Theta-rbm: Unfactored gated restricted boltzmann machine for rotation-invariant representations," *arXiv preprint arXiv:1606.08805*, 2016.
- [37] D. C. Mocanu, H. B. Ammar, L. Puig, E. Eaton, and A. Liotta, "Estimating 3d trajectories from 2d projections via disjunctive factored four-way conditional restricted boltzmann machines," *arXiv preprint arXiv:1604.05865*, 2016.
- [38] J. Gao, J. Yang, G. Wang, and M. Li, "A novel feature extraction method for scene recognition based on centered convolutional restricted boltzmann machines," *Neurocomputing*, vol. 11, no. 2, pp. p14–19, 2016.
- [39] N. Phan, D. Dou, B. Piniewski, and D. Kil, "Social restricted boltzmann machine: Human behavior prediction in health social networks," in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2015, pp. 424–431.
- [40] G. Li, L. Deng, Y. Xu, C. Wen, W. Wang, J. Pei, and L. Shi, "Temperature based restricted boltzmann machines," *Scientific Reports*, vol. 6, p. 19133, 2016.
- [41] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Learning deep hierarchical visual feature coding," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 12, pp. 2212–2225, 2014.
- [42] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *ICML*, vol. 1, 2001, pp. 577–584.
- [43] S. S. Rangapuram and M. Hein, "Constrained 1-spectral clustering," *arXiv preprint arXiv:1505.06485*, 2015.

- [44] Y. J. XIAO Yu, "Semi-supervised clustering based on affinity propagation algorithm," *Journal of Software*, vol. 19, no. 11, pp. 2803–2813, 2008.
- [45] S. Osindero and G. E. Hinton, "Modeling image patches with a directed hierarchy of markov random fields," in *Advances in Neural Information Processing Systems*, 2008, pp. 1121–1128.
- [46] J. M. Tomczak, "Learning informative features from restricted boltzmann machines," *Neural Processing Letters*, vol. 44, no. 3, pp. 1–16, 2015.
- [47] J. Zhang, S. Ding, N. Zhang, and Z. Shi, "Incremental extreme learning machine based on deep feature embedded," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 1, pp. 111–120, 2016.
- [48] I. Sutskever and G. E. Hinton, "Learning multilevel distributed representations for high-dimensional sequences," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 548–555.
- [49] V. Mnih, H. Larochelle, and G. E. Hinton, "Conditional restricted boltzmann machines for structured output prediction," *arXiv preprint arXiv:1202.3748*, 2012.
- [50] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
- [51] K. Cho, A. Ilin, and T. Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *Artificial Neural Networks and Machine Learning-ICANN 2011*. Springer, 2011, pp. 10–17.
- [52] K. H. Cho, T. Raiko, and A. Ilin, "Gaussian-bernoulli deep boltzmann machine," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–7.
- [53] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *The Journal of Machine Learning Research*, vol. 12, pp. 1025–1068, 2011.
- [54] J. Zhang, G. Tian, Y. Mu, and W. Fan, "Supervised deep learning with auxiliary networks," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 353–361.
- [55] G. Chen, "Deep transductive semi-supervised maximum margin clustering," *arXiv preprint arXiv:1501.06237*, 2015.
- [56] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [57] Y. Freund and D. Haussler, *Unsupervised learning of distributions of binary vectors using two layer networks*. Computer Research Laboratory [University of California, Santa Cruz], 1994.
- [58] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, 2010.
- [59] R. Salakhutdinov, "Learning deep generative models," Ph.D. dissertation, University of Toronto, 2009.
- [60] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.
- [61] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.
- [62] R. Karakida, M. Okada, and S. I. Amari, "Dynamical analysis of contrastive divergence learning: Restricted boltzmann machines with gaussian visible units," *Neural Networks the Official Journal of the International Neural Network Society*, vol. 79, no. C, pp. 78–87, 2016.
- [63] H. Li, M. Wang, and X.-S. Hua, "Msra-mm 2.0: A large-scale web multimedia dataset," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 2009, pp. 164–169.
- [64] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [65] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
- [66] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 126–135.
- [67] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.



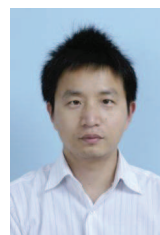
Jielei Chu received the B.S. degree from Southwest Jiaotong University, Chengdu, China in 2008, and is currently working toward the Ph.D. degree at Southwest Jiaotong University. His research interests are machine learning, data mining, semi-supervised learning and ensemble learning.



Hongjun Wang received his Ph.D. degree in computer science from Sichuan University of China in 2009. He is currently Associate Professor of the Key Lab of Cloud Computing and Intelligent Techniques in Southwest Jiaotong University. His research interests are machine learning, data mining and ensemble learning. He published over 30 research papers in journals and conferences and he is a member of ACM and CCF. He has been a reviewer for several academic journals.



Meng Hua received his Ph.D. degree in mathematics from Sichuan University of China in 2010. His research interests include belief revision, reasoning with uncertainty, machine learning, general topology.



Peng Jin received his BS, MS and Ph.D. in Computing Science from the Zhongyuan University of Technology, Nanjing University of Science and Technology, Peking University respectively. From October 2007 to April 2008, he was a visiting student at the department of Informatics, University of Sussex (Funded by China Scholarship Council); from August 2014 to February 2015, he is a visiting research fellow at the department of Informatics, University of Sussex. Now, he is a professor at Leshan Normal University (School of Computer Science). His research interests include natural language processing, information retrieval and machine learning.



Tianrui Li (SM'11) received the B.S., M.S., and Ph.D. degrees in traffic information processing and control from Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2002, respectively. He was a Post-Doctoral Researcher with Belgian Nuclear Research Centre, Mol, Belgium, from 2005 to 2006, and a Visiting Professor with Hasselt University, Hasselt, Belgium, in 2008; University of Technology, Sydney, Australia, in 2009; and University of Regina, Regina, Canada, in 2014. He is currently a Professor and the Director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or co-authored over 150 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing, and rough sets. Dr. Li is a fellow of the International Rough Set Society.